Sistema de voto eletrônico utilizando a blockchain

Henrique Niwa, Celso Mendes

[1]Instituto Nacional de Pesquisas Espaciais - INPE
Laboratório Associado de Computação e Matemática Aplicada - LABAC
Av. dos Astronautas, 1758, Jardim da Granja, CEP:12227-010
São José dos Campos - SP - Brasil

{henrique.niwa,celso}@inpe.br

Resumo. *Este trabalho mostra o estudo e implementação de um sistema de voto eletrônico, utilizando-se de blockchain, um banco de dados descentralizado e criptografado. O sistema proposto, além de oferecer ainda mais segurança ao processo de votação, permitiria uma completa auditoria na eleição, tanto do código fonte utilizado quanto da base de dados, cada eleitor poderia verificar o seu próprio voto. O desafio envolvendo computação de alto desempenho foi distribuir, simular o processo eleitoral e apurar a votação na escala de dezenas de milhões de eleitores, assim como o modelo atual em vigência no Brasil, incluindo criptografia nas transações e total transparência na apuração.*

Abstract. *This paper shows the study and implementation of an electronic voting system using blockchain, a decentralized and encrypted database. The proposed system, in addition to offering even more security to the voting process, would allow a complete audit of the election of both the source code used and the database, each voter could verify his own vote. The challenge involving high-performance computing was to distribute, simulate the electoral process, and to count the voting of tens of millions of voters, with a performance in the same level of the current model in place in Brazil, including transaction encryption and full transparency in polling.*

1. Introdução

Existem diferentes meios de votação pelo mundo, o mais comum e simples é o que se utiliza de cédulas de votação, um processo que consiste no eleitor ir a um centro designado, criar a marcação de sua preferência sendo observado por um grupo de auditores, mas com o voto ainda secreto e depositar sua ficha de papel em uma urna. Esses votos então são agregados de todos os diferentes centros de votação e posteriormente validados e contabilizados se utilizando de métodos manuais e automáticos. Para uma pequena quantidade de votantes é um meio simples e razoavelmente rápido de votar e contar, porém para grandes populações há um grande trabalho a ser feito e portanto levam-se dias para finalizar o evento. Também existe o risco de que contagens erradas, fraude nas cédulas e urnas de votação e ausência de eleitores atrapalhe e/ou mude o resultado. No Brasil temos um sistema de voto onde se utilizam urnas eletrônicas, que fazem a contagem e contabilidade local dos votos, ainda assim é necessário que as unidades de memória de cada máquina sejam enviadas para um local central e efetuada a leitura e agregação de votos, essas unidades de memória e as máquinas em si necessitam de segurança para que os dados não sejam modificados por partes maliciosas interessadas.

Para testes de performance do sistema foram levados em conta os números de eleitores do Brasil, em 2018[1] eram 146 milhões de eleitores esperados, porém na votação final foram contabilizados 116 milhões[2](figura 1), havendo 11 milhões de brancos e nulos, portanto quase 20 milhões de abstenções. Em 2018 as eleições[3] ocorreram em 7 de outubro, o prazo para troca[4] de zona eleitoral em 9 de maio e o pedido para voto em trânsito[5] em 23 de agosto, ou seja a população teve menos de 1 mês e meio para poder requisitar o direito ao voto sem estar em sua zona eleitoral de registro ou ainda 5 meses para poder votar nos representantes de sua localidade atual. Com o sistema proposto a localidade dos votos é irrelevante pois o eleitor poderá votar em qualquer dispositivo ou ainda em qualquer zona eleitoral. Essa desburocratização do voto pode mudar o cenário das eleições de acordo com a figura 2 a parcela de eleitores que poderia ser beneficiada é significativa com 20% do total. E o tempo de apuração em 2018 para o segundo turno foi iniciado a partir das 19h até a meia noite, 5 horas no total[6]. A suposição de que todas as abstenções seriam suprimidas com uma votação eletrônica baseada em blockchain é uma visão otimista, devendo haver uma coordenação entre as áreas da sociologia e modificação das regras eleitorais para tanto.

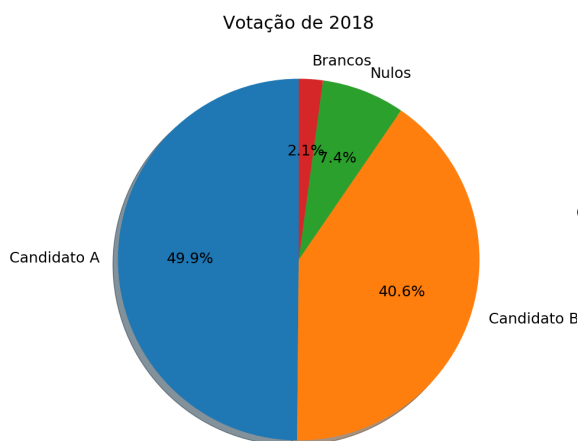
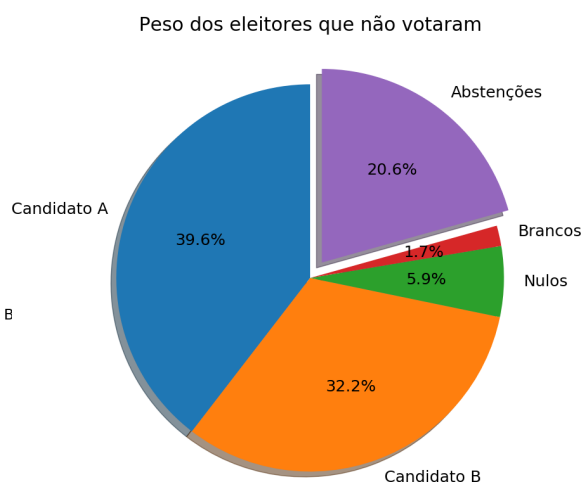
Figura 1. Votos válidos de 2018.

Fonte: Jornal Gazeta do Povo[1]

Figura 2. Parcela de eleitores que poderiam votar no sistema proposto.

Fonte: CNM - Confederação Nacional de Municípios[2]

[1] https://www.cnm.org.br/cms/biblioteca/Eleitorado-2018.pdf

[2] https://especiais.gazetadopovo.com.br/eleicoes/2018/resultados/brasil-2turno-presidente/

[3] https://g1.globo.com/politica/eleicoes/2018/noticia/eleicoes-2018-datas.ghtml

[4] https://exame.abril.com.br/brasil/prazo-para-pedir-ou-transferir-titulo-eleitoral-

[5] https://g1.globo.com/politica/eleicoes/2018/noticia/2018/08/22/termina-nesta-quinta-feira-prazo-para-eleitor-pedir-voto-em-transito.ghtml

[6] https://g1.globo.com/politica/eleicoes/2018/apuracao/presidente.ghtml

2. Metodologia

O sistema implementado buscou alguns objetivos:

1. Anonimato
2. Segurança
3. Imutabilidade
4. Acesso Remoto
5. Verificável pelo usuário
6. Auditável
7. Código aberto

O anonimato foi garantido pelo fato dos usuários terem apenas informações das chaves públicas (através dos endereços no *blockchain*) registradas no banco de dados. A segurança se teve ao utilizar criptografia estabelecida e implementações já testadas. Imutabilidade vem da própria natureza do *blockchain*, onde cada novo bloco aumenta ainda mais a segurança dos dados. Acesso remoto se teve pelo uso de uma requisição HTTP aos serviços, qualquer cliente pode ser escrito que possua uma biblioteca de requisições web. Podem ser criados clientes para computadores desktop, aplicativos para Android e iOS, navegadores web. Esta requisição utiliza JSON e poderia ser feita utilizando HTTPS assim evitando que os dados da rede sejam interceptados e se crie uma associação da chave pública com um endereço da internet e consequentemente com uma pessoa física. Verificação através do id da transação, cada voto irá retornar um identificador que irá permitir saber o destino do voto, quantidades e data. Auditabilidade completa, no final da divulgação dos resultados, é distribuído o banco de dados com todos os votos, todos poderão ver os votos, a associação entre um eleitor e sua identidade nos registros só poderá ser feito por cada um. Ainda enquanto a votação estiver sendo realizada, um nó pode ser gerido por uma auditoria, que poderá conferir em tempo real os votos. O uso de código aberto foi primordial, pois a pesquisa utilizou de verba pública, também houve cuidado ao se escolher a licença de *copyright*, pois o código original do *bitcoin* é uma licença MIT, quer dizer que o código pode ser modificado e utilizado como quiser, sem necessitar de divulgação. A licença utilizada foi a GPLv3, ela dita que qualquer modificação deve ter seus fontes divulgados.

Foi utilizado containers docker para criação de ambientes de compilação e testes, isto foi de acordo com a licença, pois segundo a GPLv3 não se deve apenas divulgar o código fonte, mas também todo o necessário para que seja feita a compilação do mesmo. Não basta disponibilizar o código-fonte, precisam ser indicadas quais versões das bibliotecas foram utilizadas e se as mesmas foram modificadas, incluir o código fonte delas. Todo o ferramental de compilação também deve ser mencionado. A praticidade de incluir uma imagem docker garante que os resultados possam ser reproduzidos rapidamente.

2.1. Avaliando propostas de voto com *blockchain* já existentes

Na literatura existem diversos trabalhos relacionados a voto e blockchain, em [Bistarelli et al. 2017] foi criado um sistema utilizando a *blockchain* pública do *bitcoin*. Primeiro há o problema do número de transações possíveis, o algoritmo do *bitcoin* foi criado de modo que a criaçã de novos blocos se faça a cada 10 minutos[Nakamoto 2009], cada bloco contém em média 2000 transações[7]. O número de votos possíveis por dia

[7] `https://outputs.today/` em 08/05/2019

seria:

$$24 * 6 * 2000 = 288.000$$

Levando em conta que as transações que não pagam taxas podem nunca ser inclusas e a taxa média [8] esta em 1,7 US$ este sistema não seria viável para grandes quantidades de votos. Dessa forma os *blockchains* públicos do *bitcoin* e do *ethereum*, não são dimensionáveis para votações de uma grande população. Seja pelo número de votos possíveis diários (*bitcoin* em 380 mil e *ethereum* 900 mil)[9] ou pelas taxas de cada transação (bitcoin 2,8 USD e ethereum 0,14 USD)[10]. Nesta implementação[11] em linguagem Go o autor não implementa verificação da autencidade das transações. No trabalho de [Hjalmarsson et al. 2018],[Cooley et al. 2018], [Shahzad and Crowcroft 2019] e [Wu and Yang 2018] são falado sobre os diversos aspectos de infraestrutura de uma eleição utilizando blockchain, mas não há implementação ou resultados de simulação. Em [Adiputra et al. 2018] e [Singh and Chatterjee 2018] são feitas propostas e parece ter havido uma implementação, mas não há resultados quantitativos. Os trabalhos de [Zhang et al. 2018], [Khoury et al. 2018]", [Yavuz et al. 2018], [M.C 2017], foram feitos com base no *blockchain* público do Ethereum, utilizando da plataforma pública e dos contratos nativos, não há resultados quantitativos. A escalabilidade da plataforma nas últimas semanas não ultrapassou 750 mil transações diárias[12]. Cada transação não pode indicar o voto de mais do que um eleitor, mesmo com os contratos inteligentes, isto torna inviável o uso para grandes populações.

2.2. Avaliando tecnologias de *blockchain* existentes

- BigChainDB[13]: um sistema que partiu de bancos de dados distribuídos e depois atribuiu características do blockchain.
- Chain Core[14] + Stellar: Uma plataforma de blockchain como serviço, o uso de sua infraestrutura é paga.
- Corda[15]: uma *blockchain* desenvolvida para uso corporativo, com uma versão aberta.
- Credits[16]: uma plataforma de desenvolvimento corporativa.
- Elements Blockchain Platform[17]: extensões para o protocolo do *bitcoin*.
- Eris:db[18]: extensões para o protocolo do *bitcoin*.
- Ethereum[19]: uma plataforma descentralizada que permite o uso de contratos dentro de sua própria *blockchain*.
- Quorum[20]: uma plataforma corporativa baseada no *Ethereum*.

[8] `https://bitinfocharts.com/comparison/bitcoin-transactionfees.html#3m` em 08/05/2019
[9] `https://bitinfocharts.com/comparison/transactions-btc-eth.html#3m`
[10] `https://bitinfocharts.com/comparison/transactions-btc-eth.html#3m`
[11] `https://github.com/codegoalie/votechain` em 08/05/2019
[12] `https://etherscan.io/chart/tx` em 05/2019
[13] https://www.bigchaindb.com
[14] https://chain.com
[15] https://www.r3.com/corda-enterprise
[16] https://credits.com
[17] https://elementsproject.org
[18] https://erisindustries.com
[19] https://www.ethereum.org
[20] https://www.goquorum.com

- Multichain[21]: uma extensão de código aberto criada a partir do código do *bitcoin*, projetada para transações financeiras de múltiplos ativos.
- Bitcoin[22]: Foi a criptomoeda inicial, criada em 2008, possui código livre e dezenas de desenvolvedores e milhões de usuários.
- Openchain[23]: uma plataforma corporativa para gerenciar ativos.
- HydraChain[24]: uma extensão do Ethereum para criação de blockchains com permissões.
- Hyperledger Fabric[25]: uma *blockchain* criada pela IBM, ela possui permissões e permite a execução de contratos inteligentes análogos ao *ethereum*.

Ainda existem plataformas de desenvolvimento criadas por grandes empresas que visam facilitar o desenvolvimento, entretanto as soluções ficam atreladas aquela empresa. IBM[26], Oracle[27], Microsoft[28], Amazon[29] possuem serviços semelhantes, cujo uso requer infraestrutura paga.

3. Desenvolvimento

A escolha deste trabalho foi inicialmente criar um código do zero, porém isto foi feito apenas de forma didática para aprender os conceitos. Em seguida se baseou no código do *bitcoin*, este projeto foi o que iniciou a revolução do *blockchain*, desde sua criação não houve nenhuma falha de segurança grave, é a criptomoeda com maior capitalização (US$ 141 bilhões), valor individual (US$ 7 mil), maior valor mediano de transferências (US$ 509) e maior número de endereços ativos (772 mil). Essa importância do *bitcoin* significa que em qualquer momento existem dezenas de milhares de desenvolvedores e usuários procurando falhas em sua rede de modo a obter acesso ou criar *bitcoins* sem efetuar os cálculos necessários. O desafio do trabalho foi escalonar a operação, a rede do *bitcoin* foi projetada desde o princípio para gerar novos blocos a cada 10 minutos3, com cada bloco tendo em média 2000 transações, o algoritmo se adapta ao poder computacional da rede fazendo com que este intervalo se mantenha não importando o número de nodos da rede. Também conta com diversas funções, distribuição dos blocos, distribuição das transações, banco de dados otimizado para gravação e recuperação das informações, criação de uma rede independente, código criptográfico robusto. Também a cada novo desenvolvimento do código principal do *bitcoin*, as mudanças podem ser incorporadas retroativamente, não sendo necessária uma equipe de desenvolvimento própria e se utilizando do conhecimento da comunidade.

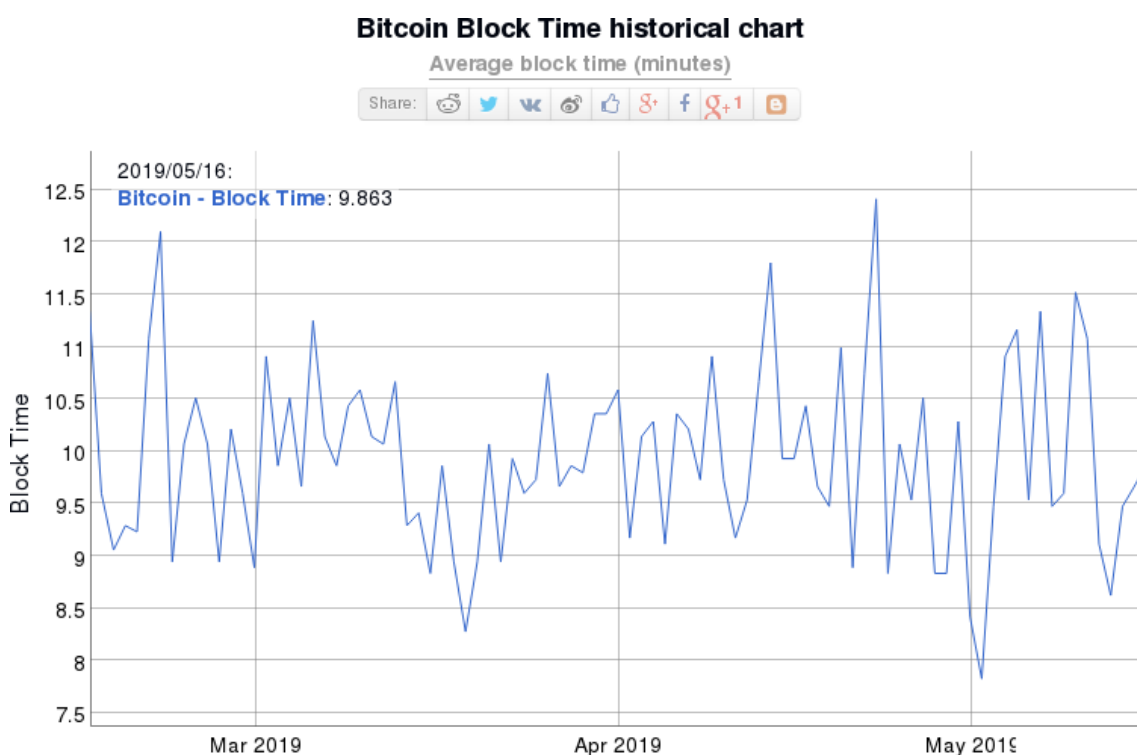
A ideia original com o código foi modificar a dificuldade mínima de cada bloco, permitindo uma geração mais rápida. Cada bloco gerado, ou minerado, recebe como prêmio pela participação na rede 50 *bitcoins*, cada moeda pode ser dividida em 100 milhões de unidades. Também modificando o intervalo e o tamanho dos blocos, permitindo uma geração mais rápida de blocos maiores. Os mineradores iriam distribuir frações

[21]https://www.multichain.com
[22]https://bitcoin.org
[23]https://www.openchain.org/
[24]https://github.com/HydraChain/hydrachain
[25]https://www.hyperledger.org/projects/fabric
[26]https://www.ibm.com/blockchain
[27]https://www.oracle.com/br/cloud/blockchain/
[28]https://azure.microsoft.com/en-us/services/blockchain-service/
[29]https://aws.amazon.com/pt/blockchain/

dos *bitcoins* para cada eleitor, processo o qual também gera transações precisando de um período para criação destes novos blocos. Qualquer nodo externo poderia se conectar a rede e incluir novos blocos, gerando para si *bitcoins* e podendo distribui-los, isso seria o equivalente a forja de votos. Isto poderia ser mitigado pela criação de uma rede de computadores protegida por *firewall*. O conceito do sistema de voto com *blockchain*, são servidores funcionando como nodos completos, que gerariam blocos e receberiam transações. No período de preparação estes servidores gerariam um par de chaves e um endereço para cada eleitor, fariam a criação das cédulas distribuindo os *bitcoins* nestes endereços. Essa associação eleitor e endereço seria de responsabilidade do governo, para distribuição das chaves uma autenticação é necessária, podendo ser uma senha cadastrada previamente ou até biometria. O problema desta ideia é que os recursos computacionais exigidos seriam cada vez maiores, também há o fato de que a maioria dos cálculos seria desperdiçado, resultando em tempo e energia gastos sem utilidade. O projeto final trabalhou modificando o tamanho dos blocos, o tempo de processamento entre cada bloco, a criação de transações especiais que criam *tokens* utilizando-os como cédulas, permissões de acesso, paralelização da apuração e da simulação de milhares de votos simultâneos. Estas modificações foram feitas com base no código do *bitcoin* e do *multichain*.

Figura 3. Tempo de geração dos blocos no *blockchain* do *bitcoin*.

Bitcoin Block Time historical chart
Average block time (minutes)
2019/05/16:
Bitcoin - Block Time: 9.863
Block Time
Mar 2019
Apr 2019
May 2019

Fonte: `https://bitinfocharts.com/comparison/bitcoin-confirmationtime.html`

3.1. Utilizando permissões de acesso

Um *blockchain* com permissões possui um controle de acesso, que dita quem pode administrar, conectar, enviar, receber, minerar. Algumas soluções já existem para isso como o Corda[Brown R. G. 2016], *Chain Core, Credits, HydraChain, BigchainDB*. Foi escolhida

a implementação do *multichain* por ser baseado e manter compatibilidade com o código do *bitcoin*, implementando este controle de acesso. Também foi escolhido pela licença do código ser aberta e qualquer trabalho derivativo também precisar ser aberto, isto concorda com os valores de pesquisa financiada com verba pública também ser aberta a todos.

O administrador possui as chaves iniciais, assim que se inicia uma blockchain, o software cria uma chave que assina os blocos e o identifica ao conectar com outros nodos. Esta chave concede permissões a qualquer outra chave. No sistema implementado as permissões padrões do são:

- Apenas nodos permitidos podem se conectar
- Apenas nodos permitidos podem criar blocos
- Todos os nodos podem enviar e receber transações
- Apenas transações permitidas podem ser enviadas
- Apenas nodos permitidos podem criar transações iniciais

Estas permissões são armazenadas dentro da própria *blockchain*, o nodo que tiver acesso e se conectar irá efetuar uma cópia de todos os blocos, transações e permissões.

3.2. Criando moedas não nativas dentro do blockchain

A criação de bens, tokens ou moedas não nativas dentro do *blockchain* é uma ideia que extende o conceito original, as transações não são mais das mesmas unidades, diferentes moedas podem ser trocadas. Um sistema de registro de uma casa de câmbio pode especificar que houve troca de dólares por euros entre dois usuários (identificados no *blockchain* pelas suas chaves). A cada eleição pode ser criada uma nova moeda e ser usada como cédula, essa cédula pode identificar o ano e local da eleição. Também permite que diferentes votações sejam realizadas ao mesmo tempo utilizando o mesmo *blockchain*.

Algumas[30] implementações[31] de blockchain oferecem a criação destes tokens. E entre elas o *multichain*, facilitando assim o reuso de código e a integração.

3.3. Segurança

Há vários níveis de segurança, onde cada um reforça o anterior. O primeiro deles é o acesso aos nodos através de uma requisição HTTP-RPC, esta possui uma senha, que pode ser divulgada aos clientes somente no dia desejado, isso evitaria que os nodos recebam requisições agentes externos. O segundo é a possibilidade de utilizar HTTPS, isto evitaria que os dados das transações sejam capturados, isto apenas divulgaria a informação de voto daquele cliente, caso a rede de entrada dos nodos seja comprometida isto divulgaria as intenções de votos de todos, os votos são assinados com as chaves privadas dos eleitores, mesmo que o voto seja capturado, ele não pode ser alterado, o https garantiria que o voto não possa ser lido, não teria a indicação do voto. O terceiro nível é a limitação das operações dos clientes com o nodo, que seriam apenas a autenticação e envio do voto, no código original do *bitcoin* podem ser requisitadas informações sobre transações, o tamanho da *blockchain*, performance do sistema, todos os blocos podem ser acessados por essa interface. A quarta medida de segurança é a geração das cédulas pelo administrador, somente ele possui a chave com permissão para tanto, não podem haver transações utilizando outras cédulas. O quinto nível é que os eleitores somente podem enviar para os

[30] https://github.com/OpenAssets
[31] https://www.openchain.org/

candidatos, as permissões de acesso são modificadas para que a lista de eleitores possam somente enviar e os candidatos somente receber, o voto dos próprios candidatos teria de ser registrado de outra forma. A parte mais delicada do sistema seria o banco de dados com as informações de usuários e suas respectivas chaves, caso haja vazamentos uma nova lista de chaves tem de ser gerada.

3.4. Possíveis problemas

3.4.1. Sybil Attack

No *bitcoin* quando uma entidade controla mais do que 50% da rede, ela pode decidir incluir blocos sem transações legítimas. No sistema proposto isto é evitado com o governo controlando a rede. E somente as transações criadas inicialmente podem ser transmitidas, somente o administrador tem autonomia pra criação das cédulas iniciais. Pela natureza aberta da apuração dos votos, qualquer manipulação feita pelo governo com os votos pode ser detectada, cada eleitor pode verificar se seu voto foi para o candidato correto e também a apuração dos votos.

3.4.2. Consenso Bizantino

Este problema é sobre o consenso entre as informações transmitidas, cada transação recebe uma identificação que pode ser usada para verifica-la. Também há o consenso entre os nodos sobre os blocos inclusos, todos terão a maior cadeia de blocos, pelo sistema proposto ser controlado, não há competição para geração de cadeias diferentes.

3.4.3. Transação duplicada

Um mesmo eleitor pode criar duas transações com saídas diferentes e a mesma entrada, mas somente a primeira sera validada recebendo um id da transação dentro da *blockchain*. Isto é um problema no *bitcoin* pelo fato de que há várias cadeias de blocos diferentes, no sistema proposto só há uma.

3.4.4. Criação de Votos Aleatórios (Interferência Externa)

A criação de um novo bloco não tem recompensa e somente as transações permitidas são inclusas. A quantidade de cédulas é fixa e criada antes da distribuição.

3.4.5. Inclusão de Blocos Aleatórios

No *bitcoin* um nodo pode entrar na rede e gerar um bloco com a dificuldade mínima correta, a recompensa por essa geração é um número de *bitcoins* que pode ser usado. Na implementação isto é evitado pois somente nodos autorizados podem fazer parte da rede.

3.4.6. Brute-Force de Votos

Uma transação precisa de uma transação de origem e uma chave privada, esta chave privada pode tentar ser adivinhada, utilizando valores aleatórios. O custo computacional é altíssimo e a recompensa seria um único voto.

3.4.7. Criptografia

A criptografia por trás das chaves *ECDSA* é robusta e a implementação usada também foi extensivamente testada pela indústria e academia. Caso ela seja quebrada, o código pode ser trocado, esta é uma das vantagens de se ter utilizado dos projetos *bitcoin* e *multichain*, uma falha grave dessas teria resposta imediata e por ser um trabalho derivado o sistema de votação pode incorporar as mudanças facilmente.

3.4.8. Tamanho do blockchain

A base de dados com 32 milhões de votos, distribuídos e votados possui 28GB. Isto não é um problema para os servidores, chamados nodos, mas caso os eleitores precisassem ter um nodo para votar isto seria impraticável. Cada cliente, que pode ser um computador desktop, celular, página web, precisará apenas das informações das chaves privada e id da transação (sua cédula), estas informações são palavras-chaves de 58 caracteres e 36 respectivamente, a criação e assinatura da transação será local, não sendo necessário guardar os blocos.

3.5. Segurança

Há vários níveis de segurança, onde cada um reforça o anterior. O primeiro deles é o acesso aos nodos através de uma requisição HTTP-RPC, esta possui uma senha, que pode ser divulgada aos clientes somente no dia desejado, isso evitaria que os nodos recebam requisições agentes externos. O segundo é a possibilidade de utilizar HTTPS, isto evitaria que os dados das transações sejam capturados, isto apenas divulgaria a informação de voto daquele cliente, caso a rede de entrada dos nodos seja comprometida isto divulgaria as intenções de votos de todos. O terceiro nível é a limitação das operações dos clientes com o nodo, que seriam apenas a autenticação e envio do voto, no código original do *bitcoin* podem ser requisitadas informações sobre transações, o tamanho da *blockchain*, desempenho do sistema, todos os blocos podem ser acessados por essa interface. A quarta medida de segurança é a geração das cédulas pelo administrador, somente ele possui a chave com permissão para tanto, não podem haver transações utilizando outras cédulas. O quinto nível é que os eleitores somente podem enviar para os candidatos, as permissões de acesso são modificadas para que a lista de eleitores possam somente enviar e os candidatos somente receber, o voto dos próprios candidatos teria de ser registrado de outra forma. A parte mais delicada do sistema seria o banco de dados com as informações de usuários e suas respectivas chaves, caso haja vazamentos uma nova lista de chaves tem de ser gerada.

4. Resultados

Os testes foram feitos utilizando um servidor com:

- 32GB de memória RAM
- 2x Intel(R) Xeon(R) CPU E5620 @ 2.40GHz (8 núcleos físicos, 16 virtuais)
- Discos rígidos mecânicos
- Rede 100Mbit

O código utilizado como base foi do projeto bitcoin e de outro código baseado no primeiro, chamado multichain. Foram criados clientes em modo texto em python, C++ e clientes gráficos em python. Nos testes de simulação de voto, foram criadas 500 instâncias nos computadores clientes, os quais possuíam:

- 8GB de memória RAM
- Processador Intel® Core™ i7-4790
- Discos rígidos mecânicos
- Rede 100Mbit

A segunda máquina utilizada como cliente possuía:

- 12GB de memória RAM
- 2x Intel(R) Xeon(R) CPU E5620 @ 2.40GHz (8 núcleos físicos, 16 virtuais)
- Discos rígidos mecânicos
- Rede 100Mbit

5. Distribuição

A distribuição dos votos consiste primeiramente da criação de uma transação especial com a quantidade total de votos a serem utilizados. Nenhum voto fora dessa transação pode ser computado pelos nodos. A partir de uma transação de origem, a próxima deve usar a quantidade exata de votos ou incluir uma saída extra com os votos restantes e um endereço. Há limitações no tamanho máximo do número de saídas da transação, nos testes o limite padrão de 4000 saídas foi utilizado. O processo de distribuição tem uma natureza obrigatoriamente sequencial, as primeiras 3999 cédulas são distribuídas entre os eleitores e a última saída para o endereço do administrador com o total de votos menos esses 3999. Essa transação gera um identificador que é usado na próxima iteração. Isso se repete até que todos os eleitores tenham recebido e o endereço do administrador ficaria com o resto das cédulas não utilizadas. Os resultados podem ser vistos na tabela 1. E a arquitetura na figura 4.

6. Simulando os votos

Os votos foram modelados como na figura 5. Utilizando de 500 clientes simultâneos, executando cada voto individualmente.

7. Apuração

O processo de apuração é um problema massivamente paralelo, são milhões de operações independentes umas das outras, para tanto foram criadas funções otimizadas para processadores de vários núcleos. As informações das transações contém individualmente um volume de dados muito pequeno, são milhões de operações de curtíssimo tempo. O processo é paralelizado utilizando *OpenMP*, onde cada *thread* recebe um bloco com várias transações, cada bloco pode conter um número variado de transações e cada transação tem um número variável de saídas, pra cada uma dessas saídas teve de ser feita a contabilidade. O escalonamento do trabalho e do número das *threads* é dinâmico por conta dessa variação.

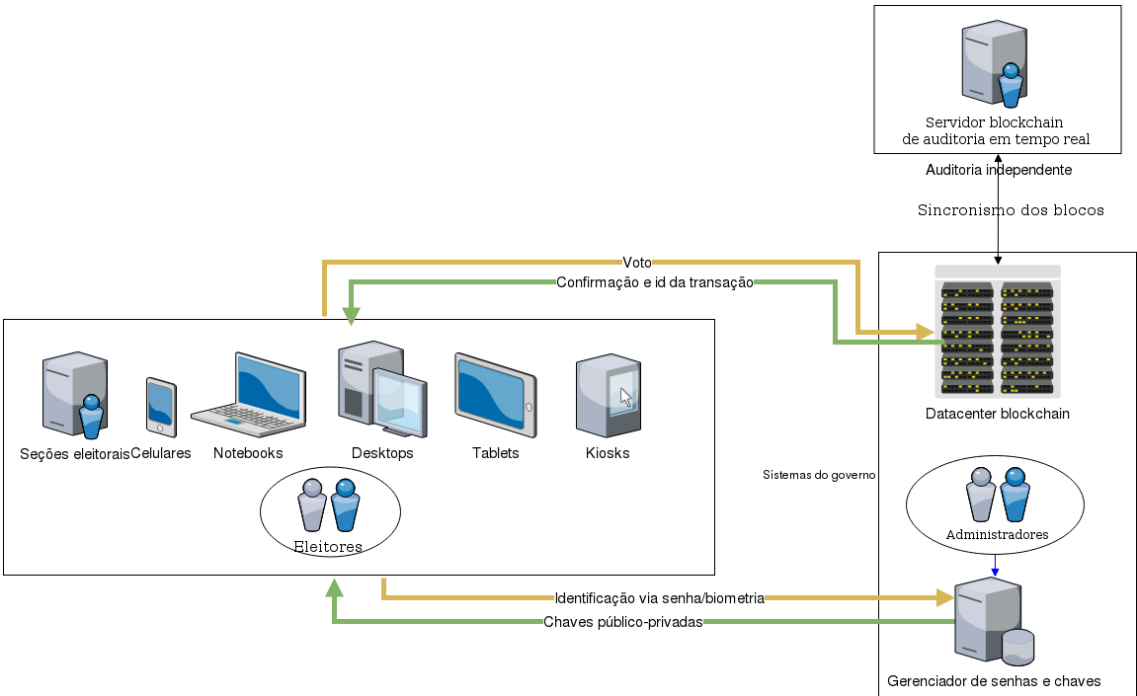
Figura 4. Arquitetura

Fonte: Autor

Tabela 1. Tempo para simulação da distribuição dos votos

Votos	Tempo	Transações
1 milhão	44,865s	251
10 milhões	509,859s	2501
100 milhões	7343,854s	25007

Tabela 2. Tamanho das bases de dados

Milhões de votos	Tamanho em MB
1	630MB
10	5400MB
35	28000MB
50	35000MB
100	55000MB

Tabela 3. Tempo para apuração de 100 milhões de votos

Número de threads	Tempo de execução
1	2929,786s
2	1480,545s
4	742,964s
8	391,053s
16	334,332s
32	406,661s

Tabela 4. Tempo para apuração com diferentes populações

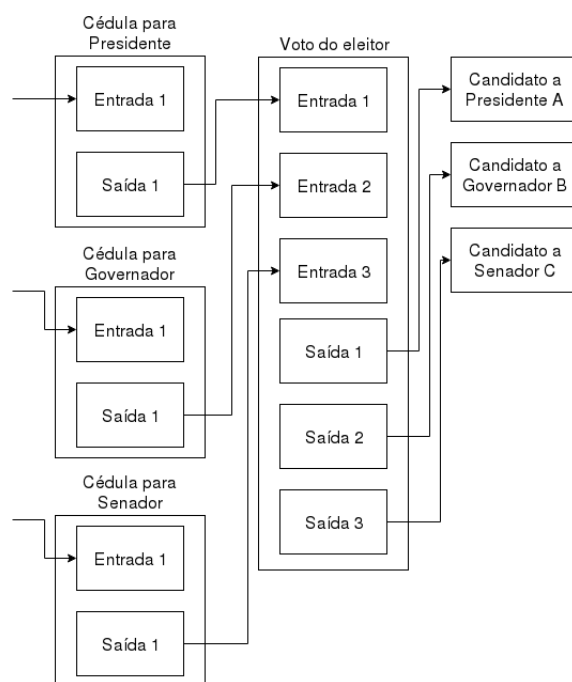
Número de votos	Tempo de execução
10 milhões	31,293s
50 milhões	176,852s
100 milhões	406,661s

Tabela 5. Tempo total da execução dos votos

Milhões de votos	Tempo de execução
50	29h37m

Tabela 6. Tempo de execução de cada voto

Mínimo	Máximo	Mediana	Média
0,709s	35,231	0,799s	1,064s

Figura 5. Modelagem de um voto como transação.

Fonte: Autor

8. Auditoria

Segundo Rivest[Rivest 2006], um dos autores da chave de criptografia RSA e autor do Princípio da Independência de Software em Sistemas Eleitorais: "Um sistema eleitoral é independente do software se uma modificação ou erro não-detectado no seu software não pode causar uma modificação ou erro indetectável no resultado da apuração." De acordo com este princípio, o presente trabalho atende, pois sua implementação pode ser feita em outra linguagem ou arquitetura desde que respeitando as regras do *blockchain*, usando as mesmas funções matemáticas, regras do protocolo e com as mesmas regras de transação. A prova disso é que existem outros clientes bitcoin em outras linguagens, para a apuração dos votos, poderiam ser implementadas as regras nestas outras linguagens que interpretem as transações como votos.

O código fonte compactado fica abaixo de 10MB, isto inclui os fontes originais do *bitcoin* e *multichain* mais modificações e melhorias implementadas pelo autor. A imagem docker do ambiente de compilação com todas as bibliotecas instaladas ocupa 600MB instalado.

Cada voto ou transação, gera um identificador (listagem 8). Ele pode ser usado para recuperar os dados de voto e assim validar se a escolha do eleitor foi mantida ou manipulada. Esse identificador pode ser convertido para um *QR code* (listagem 7) para facilidade de uso. O eleitor interessado em auditar seu próprio voto pode fazer o download do código, a base de dados, compilar ele mesmo e validar as informações, nessa transação haverá a transação origem da cédula dele, pra quais endereços de candidatos ele enviou e a quantidade de votos.

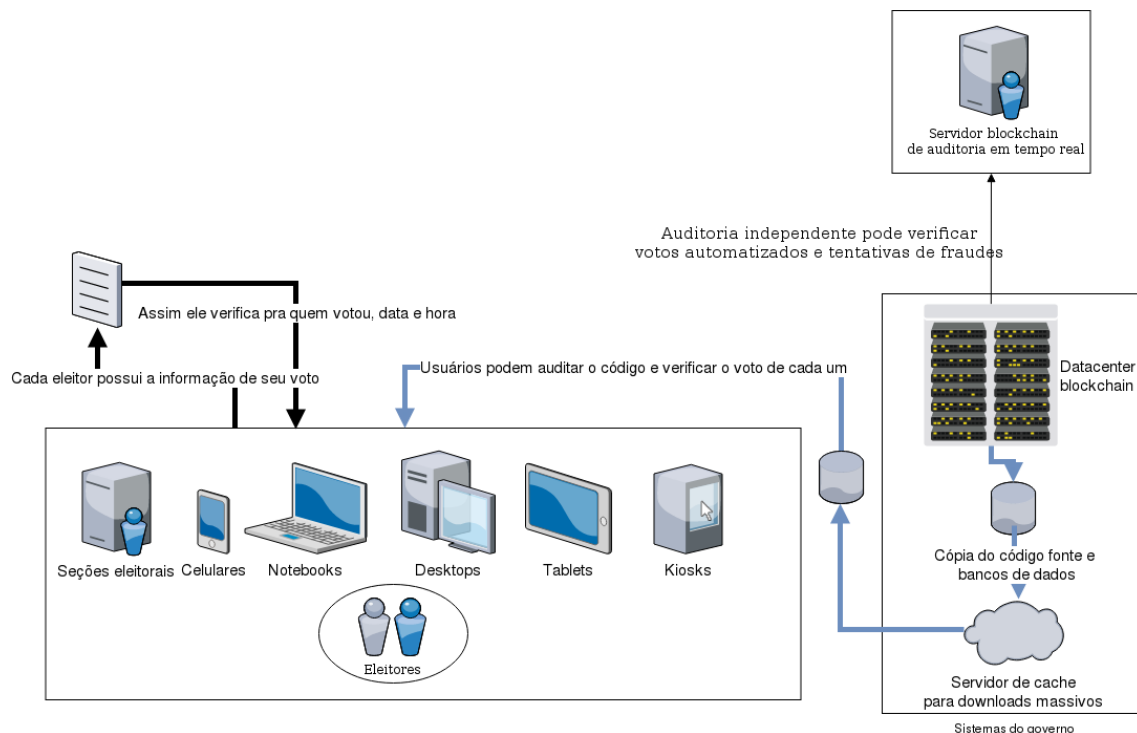
Figura 6. Processo de auditoria.

Fonte: Autor

Figura 7. QR Code verificador.

Fonte: Autor

Figura 8. Identificador da transação validada

7d2572c5426faca62f37e6ab275fafd792869e9ee2f...

Fonte: Autor

8.1. Análise dos Resultados

Pelo monitoramento da implementação temos que inicialmente o número de transações por hora era de 2 milhões, mas ao término de 24 horas haviam sido executados 43 milhões de votos, aproximadamente 10% a menos em relação ao esperado inicialmente. Esta ineficiência pode ser devido a escrita das bases de dado em disco, a parte principal do programa responsável pela escrita em disco também acabou afetando o recebimento de novos votos, com o tempo isto foi acumulando. Também temos que o consumo de memória se manteve constante por volta de 1,5GB. O tempo para recebimento e consolidação das transações em blocos pode ser melhorado, utilizando um particionamento dos eleitores

em diferentes servidores, o resultado foi com um único servidor, a vantagem de se utilizar o código do *bitcoin* é que se já tem a implementação de código para redistribuir transações e blocos, efetivamente criando uma rede distribuída. Também pode-se criar várias *blockchains* de acordo com o tamanho da população, uma por estado, ou cidade, até mesmo bairro. A indicação de para qual nodo o eleitor seria direcionado ficaria logo após a identificação e envio da chave privada e *txid* da cédula. Na apuração o processo é feito em paralelo entre as várias blockchains regionais, nelas os resultados serão ainda mais rápidos pelo menor número de blocos em cada servidor. Em comparação com as 5 horas do sistema atual, o ganho foi considerável, onde um único servidor conseguiu apurar de forma correta 100 milhões de votos em pouco mais de 400 segundos. Os tempos para execução do código também são baixos, onde a mediana ficou em 0,8 segundos, isto possibilitaria a execução em dispositivos com pouco poder de processamento. A distribuição de 100 milhões de cédulas levou pouco mais de 2 horas, este processo seria no pior caso, em várias *blockchains* distribuídas em estados, cidades, bairros, o processo seria mais rápido ainda.

9. Conclusões

Os resultados foram animadores, levando em conta que a *blockchain* utilizada pelo *bitcoin* teve uma média diária de 380 mil transações e a *blockchain* do *ethereum* 900 mil[32], o sistema desenvolvido obteve 43 milhões utilizando a mesma estrutura de segurança do *bitcoin* com um processador lançado em 2010[33]. Os resultados serão melhores utilizando melhor hardware, mas também há limitações no algoritmo que precisam ser verificadas, a geração e inclusão de milhões de transações precisa evitar conflitos de hash e há várias travas para o acesso às estruturas de dados, para evitar condições de corridas nas *threads* concorrentes. O acesso a disco também requer milhares de operações não sequenciais por segundo, que poderiam ser beneficiadas utilizando de armazenamento flash. A busca por resultados quantitativos utilizando as proporções de uma eleição no Brasil, provaram que o *blockchain* pode ser uma alternativa viável e superior na questão de transparência, ainda que haja vários elementos que podem ser melhorados.

Referências

Adiputra, C. K., Hjort, R., and Sato, H. (2018). A proposal of blockchain-based electronic voting system. In *2018 Second World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pages 22–27.

Bistarelli, S., Mantilacci, M., Santancini, P., and Santini, F. (2017). An end-to-end voting-system based on bitcoin. In *Proceedings of the Symposium on Applied Computing*, SAC '17, pages 1836–1841, New York, NY, USA. ACM.

Brown R. G., Carlyle J., G. I. H. M. (2016). Corda: An introduction.

Cooley, R., Wolf, S., and Borowczak, M. (2018). Blockchain-based election infrastructures. In *2018 IEEE International Smart Cities Conference (ISC2)*, pages 1–4.

[32] https://bitinfocharts.com/comparison/transactions-btc-eth.html#3m

[33] https://ark.intel.com/content/www/us/en/ark/products/47925/intel-xeon-processor-e5620-12m-cache-2-40-ghz-5-86-gt-s-intel-qpi.html

Hjalmarsson, F. P., Hreioarsson, G. K., Hamdaqa, M., and Hjalmtysson, G. (2018). Blockchain-based e-voting system. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 983–986.

Khoury, D., Kfoury, E. F., Kassem, A., and Harb, H. (2018). Decentralized voting platform based on ethereum blockchain. In *2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*, pages 1–6.

M.C, A. M. S. (2017). *Ethervoltz: um sistema de votação auditável baseado no blockchain ethereum*. Graduação em engenharia da computação, ETEP Faculdades/Faculdade de tecnologia de São José Dos Campos.

Nakamoto, S. (2009). Bitcoin: A peer-to-peer electronic cash system. *Cryptography Mailing list at https://metzdowd.com*.

Rivest, R.L; Wack, J. (2006). On the notion of "software independence" in voting systems.

Shahzad, B. and Crowcroft, J. (2019). Trustworthy electronic voting using adjusted blockchain technology. *IEEE Access*, 7:24477–24488.

Singh, A. and Chatterjee, K. (2018). Secevs : Secure electronic voting system using blockchain technology. In *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, pages 863–867.

Wu, H. and Yang, C. (2018). A blockchain-based network security mechanism for voting systems. In *2018 1st International Cognitive Cities Conference (IC3)*, pages 227–230.

Yavuz, E., Koç, A. K., Çabuk, U. C., and Dalkılıç, G. (2018). Towards secure e-voting using ethereum blockchain. In *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, pages 1–7.

Zhang, W., Yuan, Y., Hu, Y., Huang, S., Cao, S., Chopra, A., and Huang, S. (2018). A privacy-preserving voting protocol on blockchain. In *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, pages 401–408.